Experiment Software

Chris Green for the SciSoft team

FNAL AI Infrastructure Planning Mini-workshop, 2023-04-06

Overview

- AI apps/frameworks in use.
- Packaging/support considerations and consequences.
- UPS -> Spack.

AI apps/frameworks in use

Packaged by SciSoft:

- OpenCV
- libTorch (part of PyTorch)
- Tensorflow
- Triton (formerly known as `trtis-client`)

Other software is used/packaged by experiments but relies on link compatibility with the art framework with respect to compilers, language standards and dependencies. Currently known:

- larcv2
- larcv3

Packaging/support: considerations

- Many (many, many . . .) dependencies.
- Active development: “Move Fast and Break Things™.”
- Complex configuration: threading/accelerators (e.g. CUDA, OpenCL, Intel Threading Building Blocks, OpenMP, MPI) or application-specific features (often triggering more dependencies).
- Entitlement! AI packages often assume that they are/own the world: they are monolithic systems that download and build their own dependencies—usually with either Bazel or with CMake’s external project facility.
- Experiments usually utilize AI in the context of their own software ecosystem, requiring link/binary compatibility with respect to C++ and/or Fortran compilers and language standards, and shared dependencies.

Packaging/support: consequences

Building and distributing a new version of an AI package usually involves *lots* of expert work on the build scripts[1], including:

- Careful review of build system, including specification of dependencies to identify new or updated packaging or patch requirements.
- Construction and application of homespun patches and application to package source to:
  - allow for external dependencies or language standard selection
  - add compiler options or fix source for different compilers or language standards
- Post-install processing for relocatability.

[1] Tensorflow 2.3.1 -> 2.6.0 required nearly 5000 lines of script code and patches, comprising multiple FTE weeks of effort.

Packaging/support: consequences

In addition, the fast-evolving nature of AI applications means that one or more experiments might require a new major or minor version of a package while others might want only point releases or complete stability.

UPS -> Spack

- ↑ Many packages already have recipes upstream in Spack proper or in community-maintained recipe repositories.
- ↑ Relocatability is often automatic via BuildCache.
- ↑ Well-defined and widely-used domain-specific language for describing build customizations.
- ↑ Spack allows straightforward support of similar stacks with some version changes in dependencies.
- ↓ Upstream recipes are often not as general as they should be due to contributors' specific requirements $\implies$ careful review and customization will still be required.
- ↓ Customization of existing upstream recipes will require external review and more effort for generalization to avoid the burden associated with maintaining FNAL-specific recipes.